

Colloque ANR STIC - 14 décembre 2010

Projet VALMEM

(ANR-06-ARFU-005 : 01/01/2006 - 31/12/2010)

VALIDATION FONCTIONNELLE ET TEMPORELLE DES MÉMOIRES
EMBARQUÉES, DÉCRITES AU NIVEAU TRANSISTOR, PAR DES MÉTHODES
FORMELLES

E. ANDRÉ, A. BARA, P. BAZARGAN-SABET, R. CHEVALLIER,
E. ENCRENAZ, L. FRIBOURG, D. LE DÛ, P. RENAULT

LAB. INFORMATIQUE PARIS 6 - LAB. SPÉCIFICATION VÉRIFICATION
STMICROELECTRONICS

Plan

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Objectifs et organisation du projet

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Objectifs

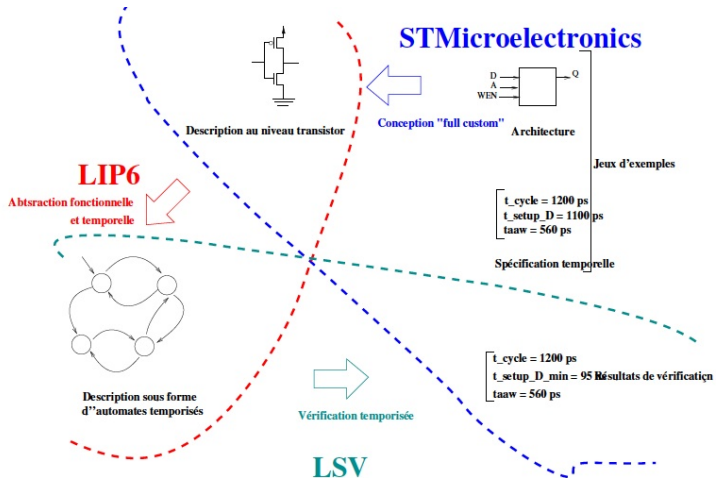
Contexte :

- Les circuits mémoires ont la particularité d'intégrer des fonctionnalités toujours plus complexes tout en devant répondre à des objectifs de performances accrues. Pour ces raisons, ces circuits sont conçus directement au **niveau transistor**, ce qui rend très difficile leur validation.

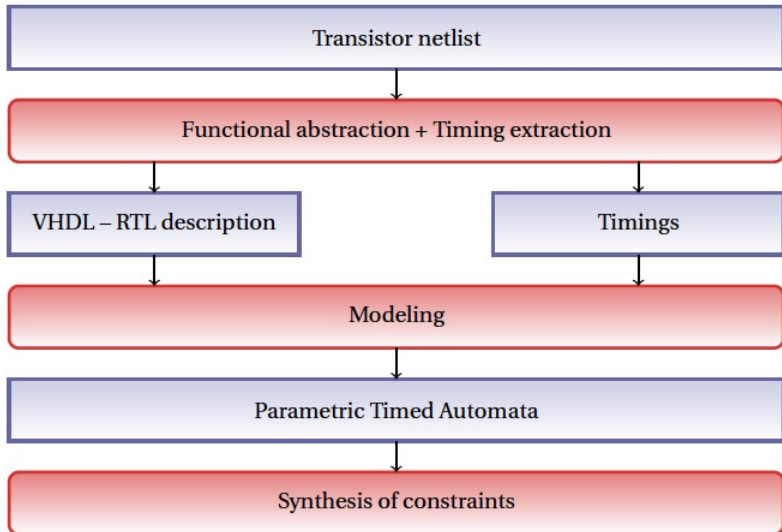
But :

- Vérification fonctionnelle et temporelle de circuits mémoires en utilisant des **méthodes formelles** (par opposition à des méthodes basées principalement sur de la simulation électrique)

Vue schématique de l'organisation VALMEM



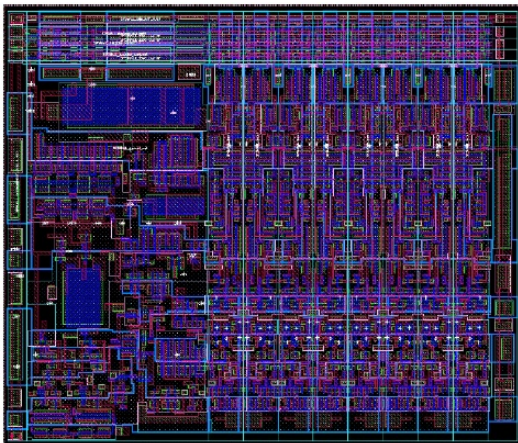
Méthodologie du projet VALMEM



Exemple de la mémoire SPSMALL

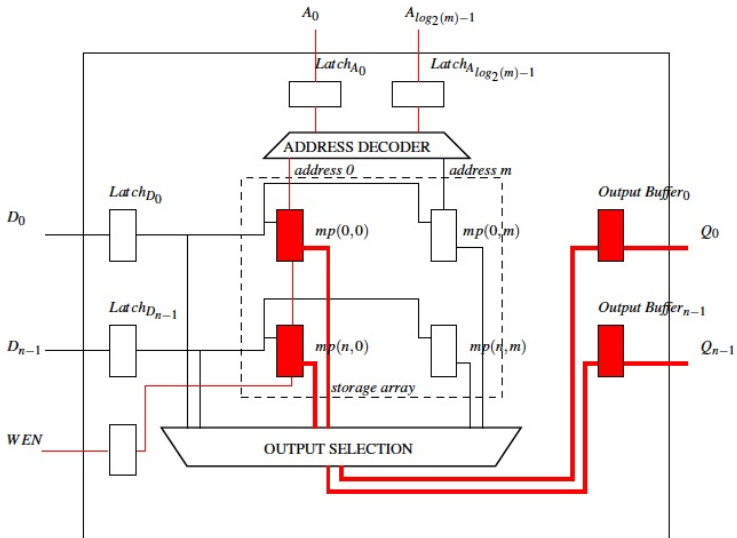
- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL**
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Exemple de la mémoire SPSMALL

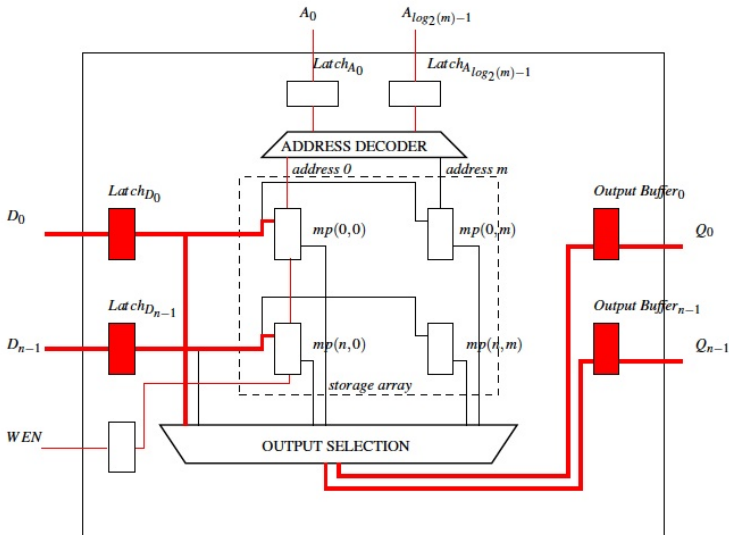


Mémoire embarquée (130 nanos) commercialisée par STM

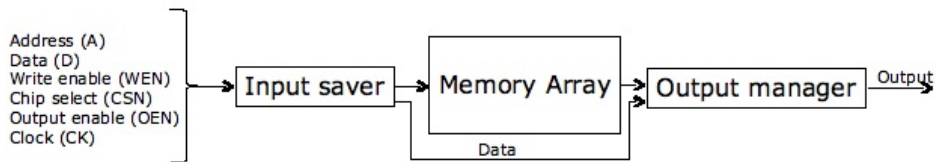
Architecture globale de SPSMALL (lecture)



Architecture globale de SPSMALL (écriture)



Vue synthétique de la mémoire SPSMALL

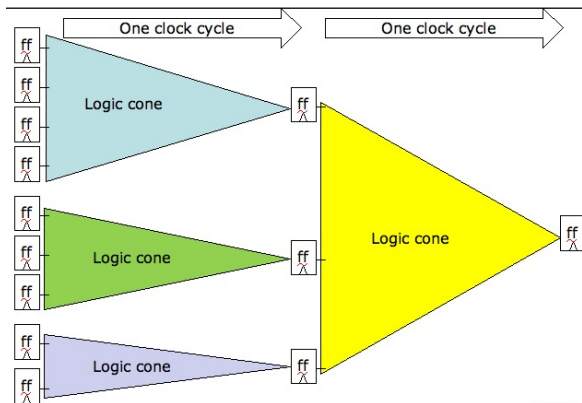


Noter l'action de *write-through* qui permet l'écriture en mémoire de façon *asynchrone*

Difficultés intrinsèques liées à l'analyse de mémoire asynchrone

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone**
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Diffic. analyse SPSMALL / circuit purt synchrone



Contrairement au schéma ci-dessus, l'analyse des temps de calcul de SPSMALL ne peut pas se faire par simple addition de temps de cycle

Autres difficultés intrinsèques à SPSMALL

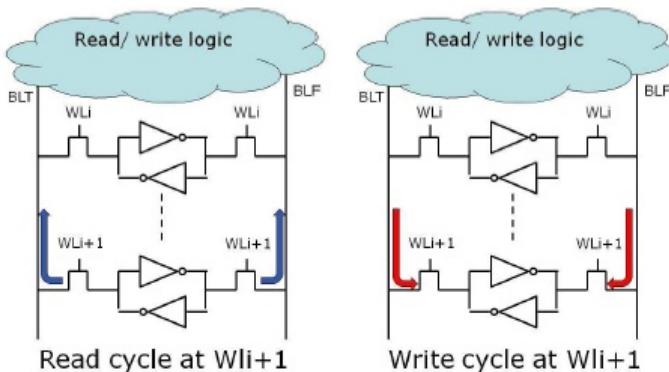
- 1 éléments d'asynchronisme (vs. circuit purement synchrone)
- 2 full custom (vs. standard cell)
- 3 usages multiples d'un même composant dans des environnements différents

Difficulté 2 : analyse SPSMALL / *standard cell*

L'abstraction fonctionnelle classique utilisant des outils à base de **pattern matching** ne marche pas dans une conception "full custom" comme la SPSMALL

↪ techniques innovantes à base de **coloriage de graphe**

Illustr. diffic. 3 (usage multiple d'un \hat{m} transistor)



Abstr. fonctionnelle et temporelle (outils MYGALE et TIMEX)

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)**
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Prototype de l'outil d'abstraction MYGALE

Rôle : construire une netlist de portes à partir description niveau transistor

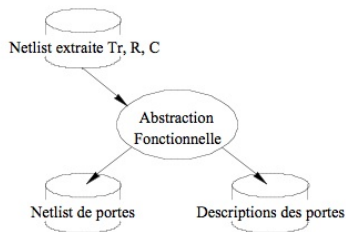


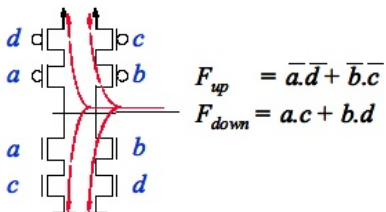
FIG.: Architecture générale de l'outil d'abstraction

Sorties outil :

- description des portes en format SPICE
- fichier format VHDL

Abstraction fonctionnelle

Méthode formelle : reconnaître la fonction d'un groupe de transistors en **traçant les chemins de courant**



L'**expression booléenne** de la porte est obtenue en établissant les conditions de mise à 1 (F_{up}), et de mise à 0 (F_{down})

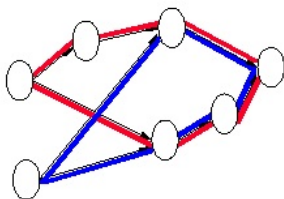
Difficultés rencontrées par abstraction fnelle

Difficulté 1 :

- Portes conflictuelles

Résolution en :

- 1 Prouvant que la condition de conflit n'est pas réalisable
- 2 Identifiant les corrélations entre les signaux par **coloriage du graphe** du circuit



Difficultés par abstraction fnelle (suite)

Difficulté 2 : Reconnaître les **points mémoire**

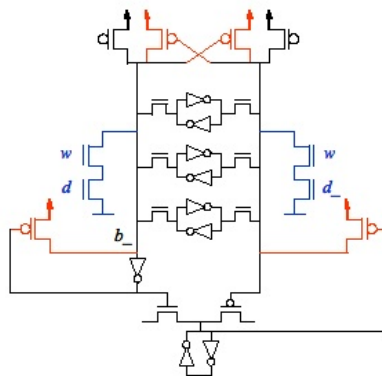
Les **dépendances cycliques** dans le graphe du circuit indiquent en général la présence de points mémoire

Ces dépendances sont représentées sous la forme d'équations booléennes

Résolution en :

- Identifiant les points mémoire par résolution des systèmes d'**équations booléennes**

Résultats abstraction fonctionnelle SPSMALL



Abstr. (y compris colonne de pts mémoire) + gén. VHDL en 11 s

Abstraction temporelle avec outil TIMEX

Objectif : obtenir les temps de propagation des portes

- Un temps de propagation par type de transition (\uparrow ou \downarrow) et par configuration des entrées
- Chaque porte modélisée par un graphe de configurations
- Temps de propagation obtenus par simulation électrique

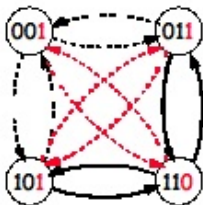


FIG.: Exemple : Nand2 – les transitions en trait plein sont caractérisées

Difficultés rencontrées par abstraction temporelle

Difficulté :

- Identifier les transitions non fonctionnelles dues aux corrélations entre les signaux

Deux types de corrélations : spatiales et temporelles

Résolution par :

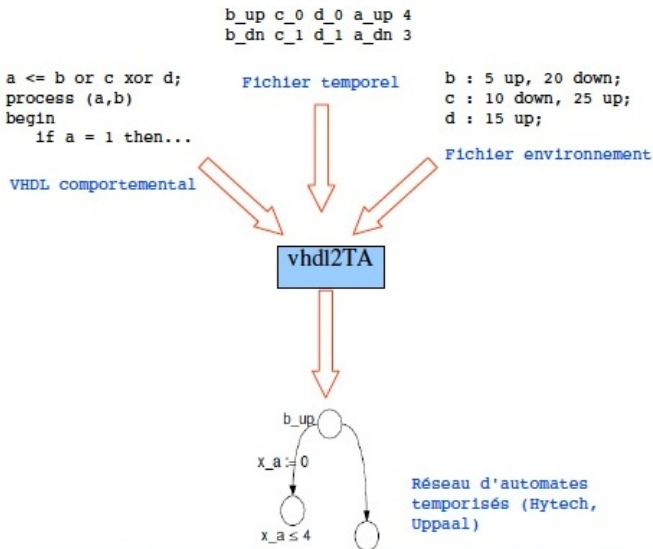
- 1 Identification des corrélations spatiales par coloration de graphe
- 2 Prise en compte des corrélations temporelles non résolues
 - ▶ corrélation des trans. pour signaux préchargés
 - ▶ corrélation provoquant la trans. de plusieurs sgnx d'entrée

Résultat : synthèse de délais pour les centaines de milliers de composants de la SPSMALL en 10h

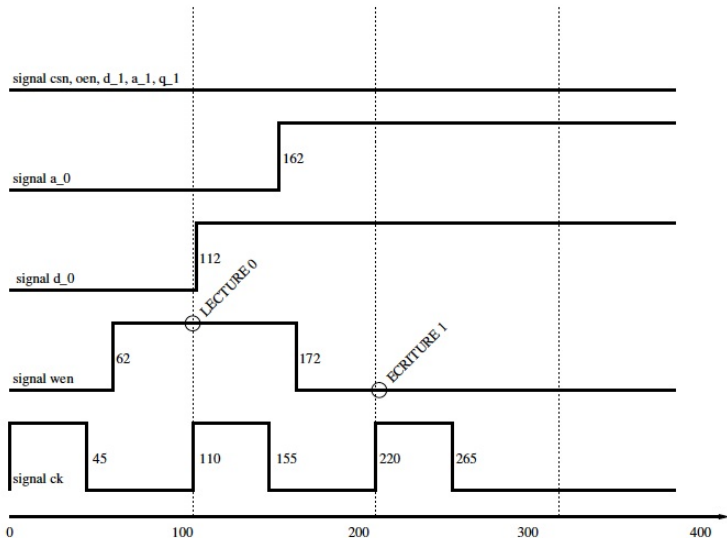
Conversion en automates temporisés (outil VHDL2A)

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)**
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Schéma de l'outil VHDL2TA

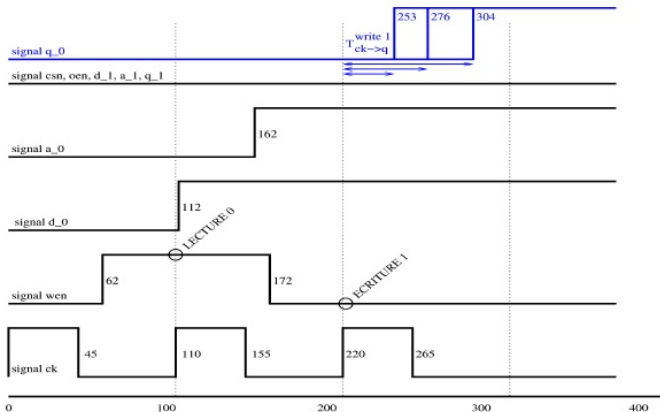


Analyse avec VHDL2TA : rappel envrt SPSMALL



Analyse SPSMALL (suite 1) : tests évolutifs envrt

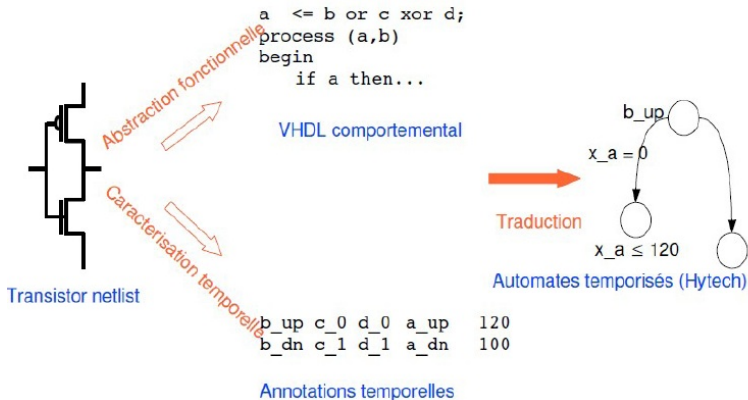
- Environnement de test :



Analyse SPSMALL avec chaîne complète outils

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils**
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions

Schéma récapitulatif de la chaîne



Analyse SPSMALL par chaîne outils VALMEM

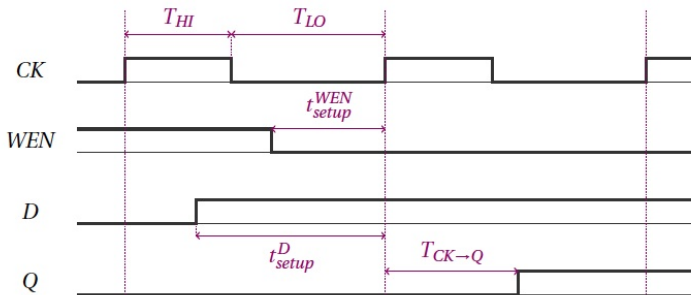
- production par **MYGALE** de VHDL (62 affectations concurrentes + 30 processus séqu. + 8 sgnx non connectés)
- production par **TIMEX** de fichiers de timings associés aux portes et latches
- conversion via **VHDL2TA** en modèle par automates temporisés (92 automates temporisés avec autant d'horloges symboliques et plusieurs centaines de délais)
- tests évolutifs avec outil de model checking UPPAAL

↪ résultats de timings globaux conformes avec la datasheet STM donnés avec une incertitude de 5%

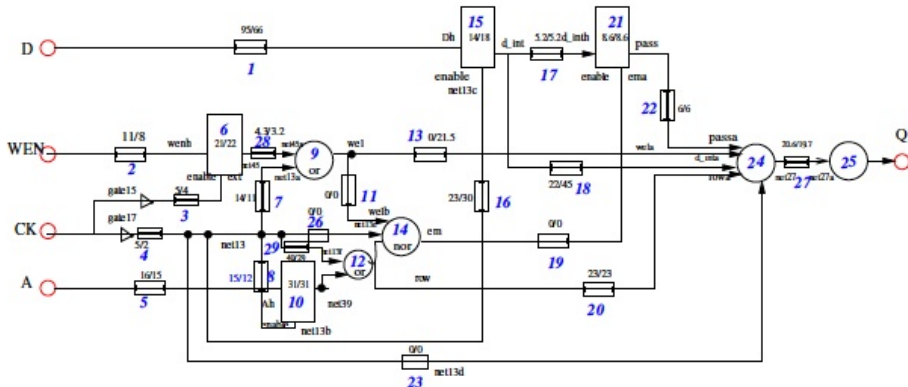
Analyse paramétrée sur modèle simplifié (outil IMITATOR)

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)**
- 8 Conclusions

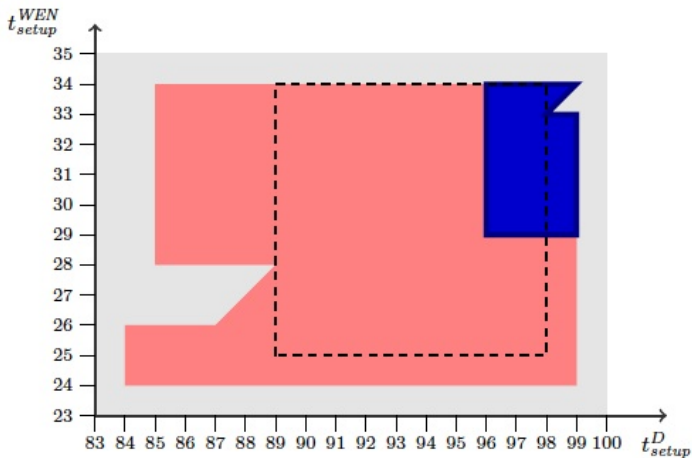
Environnement simplifié de SPSMALL



Modèle simplifié de SPSMALL



Contrainte synthétisée sur le modèle simplifié



$$\text{Contrainte : } 96 \leq t_{setup}^D \leq 98 \wedge 29 \leq t_{setup}^{WEN} \leq 34$$

Rés. présentant écart 15% / datasheet - Passage éch. réelle imposs.

Conclusions

- 1 Objectifs et organisation du projet
- 2 Exemple de la mémoire SPSMALL
- 3 Difficultés intrinsèques liées à l'analyse de mémoire asynchrone
- 4 Abstract. fonctionnelle et temporelle (outils MYGALE et TIMEX)
- 5 Conversion en automates temporisés (outil VHDL2A)
- 6 Analyse SPSMALL avec chaîne complète outils
- 7 Analyse paramétrée sur modèle simplifié (outil IMITATOR)
- 8 Conclusions**

Conclusions : Un bilan en demi-teinte

Points positifs

- 1 Abstr. fnelle : sans pattern-matching ni d'intervention humaine
- 2 Extraction des timings affinée par algo en 2 passes
- 3 Génération automatique de modèles d'automates temporisés
- 4 Synthèse de paramètres de modèles temporisés
- 5 Chaîne d'outils complète

Points négatifs

- 1 Incertitudes sur les résultats temporels obtenus (plus de 5 %) trop grandes / ceux obtenus par simu. par STM (moins de 1 %)
- 2 Résultats insuffisamment précis pour permettre la validation de la mémoire SPSMALL.
- 3 L'étude d'autres mémoires plus complexes que SPSMALL envisagée dans la proposition VALMEM n'a pas été faite.

Conclusions (suite)

Explications des points d'achoppement :

- 1 Abstraction des points mémoire par méthode formelle beaucoup plus longue et difficile qu'initialement escompté
- 2 Dispersion des tps de réponse des pts mémoire très importante lorsque ttes les configurations (m[^]m non fnelles) sont considérées
↔ nécessité d'une étape de réduction des intervalles aux seules val. fnelles. Elimination nécessite un raisnmt complexe sur la dynamique des signaux.

Finalement :

- 1 Nos outils peuvent être sans doute utilisés lors de phases de conception amnt mais pas pour valider a post. circuit full-custom
- 2 Les outils de traduction VHDL - automates puis de vérification peuvent sans doute être utilisés de façon plus pertinente pour d'autres types de circuit (synchrone, standard-cell).

Outils et publications

- Outils :

- ① MYGALE et TIMEX (page Web en cours de finalisation)

- ② VHDL2TA

- (<http://www.lsv.ens-cachan.fr/~encrenaz/valmem/vhdl2hytech/index.html>)

- ③ IMITATOR

- (<http://www.lsv.ens-cachan.fr/~andre/IMITATOR2/>)

- Publications :

- ① A. Bara, P. Bazargan, R. Chevallier, E. Encrenaz, D. Le Dû, P. Renault. [Formal verification of timed VHDL programs](#), Forum on Specification and Design Languages, 2010

- ② Etienne André. [IMITATOR II : A tool for solving the good parameters problems in timed automata](#), ENTCS 39, 2010

- ③ E. André, T. Chatain, E. Encrenaz, L. Fribourg. [An inverse method for parametric timed automata](#), IJFCS 20(5), 2009